BYTE BACK

"ByteBack"

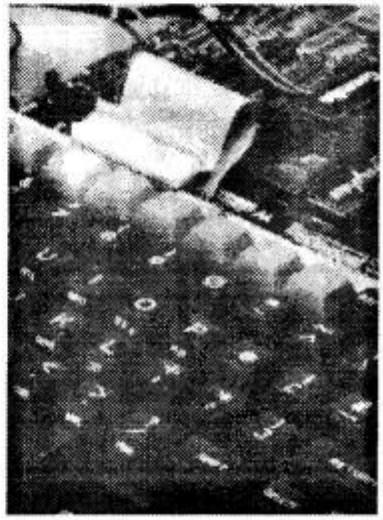
ISSUE 11 • 1995

IT'S A GAMBLE!

or how to win at the National Lottery in 37 easy steps... 12

SOFTKEYS 46

The art of writing programs in short bursts



EXTRAORDINARY

Or is it? 37

USING YOUR COMPUTER

Advice on the Beeb's various uses

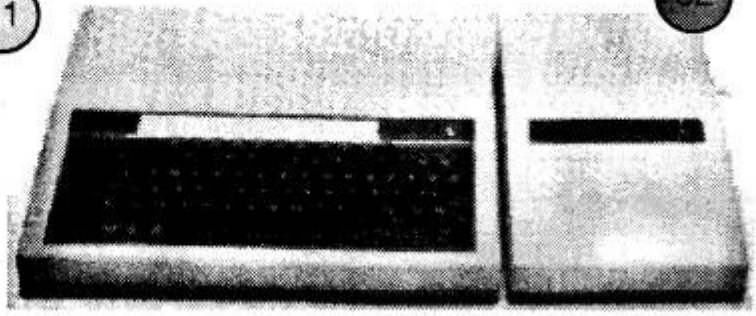
INFLATION

By 2010AD, this copy of ByteBack will be worth about £7,000

HINTS AND TIPS 11

More useful tidbits

Plus, your letters, and Classified ads 23



02

A DIFFERENT WAY OF LOOKING AT YOUR BBC MICRO

EDITORIAL

What's going on in this issue?

It's another year away from the launch of the humble Beeb, another year in which Beeb owners will shed the weight of their machine and opt for something new. It's also another year in which second hand BBC's are bought from the likes of car boot sales, and new users are born. I hope that it will also be a year in which your BBC continues to entertain and provide you with enjoyment.

Thank you to everyone who contributed to ByteBack in any way since it's beginning. Thank you for all of the letters, which I have enjoyed reading. Thank you for all of the programs, most of which have been used, and many more of which appear in this issue.

Finally, the bad news. Due to extraordinary demands on my time with work (we are having to take on somebody new at our company, just to help us cope), I will have to discontinue running ByteBack.

The company that my wife and I started three years ago, has grown. We are in excellent office accomodation, and my brother, who left the company he was working for, joined us in October. Now, just before the summer (our busiest period of the year by far), we are making plans for an additional member of staff. We are already being stretched to the limit and this has always been our quiet period.

It's not that there isn't support for the BBC anymore because there still is. It isn't that I don't have enough material to put BB together because I do. It's purely the time-restraint factor. It takes over 16 hours to fully produce each issue, including working through letters and programs, printing, folding and distributing, and all of the paperwork. I am sorry to have to let you down if you look forward to each issue: there's no other way around it for me.

I can offer an alternative to you if you are concerned that support for the Beeb now doesn't exist. Chris Richardson runs 8 Bit Software, a monthly (I don't know how he does it!), disc-based 'magazine' that has lots of letters, tips, programs, contacts, advice, utilities, and anything else BBC related. There are over 200 members in the club (myself included: K1Y), and it's a great way to stay in touch with fellow BBC users. Chris is a very nice person and would enjoy hearing from you. If you would like to receive details from 8BS, send an SAE, or for the introductory package, send 2 formatted double sided, 80 track discs (ADFS or DFS) and return postage.

Even though I won't be running ByteBack anymore, I will remain a member of 8BS, so you will find me over there from time to time.

Thank you to everyone for making ByteBack enjoyable for me. It's an end of an era as we enter this last issue, but it's not the end of the Beeb. After everything is considered, that is the reason that ByteBack ever existed.

Paul

IT'S A GAMBLE!

Bernard Beeston

So you want to win a million! Prompted by **Random Swizz!** in ByteBack Iss9, and the advent of the National Lottery I decided to write a short program to select six lottery numbers. You could do worse than use it, you certainly couldn't do better. And if your Beeb gives you a winning line, don't forget to give it a share...

This program was written and tested on BASIC 2 but should run on any of the Acorn machines using BBC BASIC, including the old BASIC 1.

HOW THE PROGRAM WORKS

Line 10: Dimensions an array of 6 variables (remember, BBC BASIC arrays include 0)

Line 20: Seeds the RND function with a truly random number and the source of this in the Beeb's TIME variable.

Line 30: Starts the loop that draws the numbers.

Line 40: Places a random number into the temp variable

Lines 50-70: Lottery numbers are not quite random as no number is called twice, which could happen if we jsut selected six random numbers in the range 1 to 49. So this nested loop checks that the next number has not already been selected. If a copy is found line 60 places a zero in temp.

Line 80: Tests for zero in temp and if so returns to line 40 to select a new number. You may think that this jump could have been done at line 60, but that would have meant jumping out of the loop. This is bad paractice and something that BBC BASIC does not like anyway.

Line 90: Places the number (now known to be unique) from temp into the next free array position.

Line 100: Completes the selection loop.

Lines 110-160: Bubble sorts the array into ascending numerical order, see BB Iss7 & 8 for details.

Line 170: Displays your winning (*sic*) line.

Line 180: Leaves a space if the program is run again.

Line 190: Speakes for itself.

To see the selected numbers before they are sorted, duplicate line 170 as line 105.

To obtain more selections just RUN the program again, or add thie line to the listing: 185 wait=GET:RUN then each time a key is pressed another selection of numbers will be displayed.

If *FX6 (RETURN) and CTRL/B are entered before the program is run the results will also be printed out – always assuming you have a printer connected.

```
10 DIM ball(5)
20 dump=RND(-TIME)
30 FOR draw=0 TO 5
40 temp=RND(49)
 50 FOR test=0 TO 5
 60 IF temp=ball(test) THEN temp=0
 70 NEXT
 80 IF temp=0 GOTO 40
 90 ball(draw)=temp
 100 NEXT
```

Program continues on page 7

LETTERS

Your thoughts and ideas

Thanks for the latest copy of ByteBack. I have found all the issues to be both interesting and instructive. Please keep going as long as possible. For what it's worth I have purchased several Public Domain disks from Chris Richardson of 8-Bit Software and generally they have been pretty good value. I can definitely endorse your recommendation to obtain the hints and tips disc BBC PD 147. My only reservation regarding other discs available, is that often there is a lack of user information which can cause difficulties if a program does not run as expected. Even listing the program doesn't always help as often the authors leave out the REM statements which they themselves used during compilation, probably to save disc space. I know Chris has often stressed to his contributors the necessity of including instructions but obviously it's out of his hands.

HELP WANTED:

Quite a few of the 8-Bit discs use the DFS format and as I prefer using ADFS I have been atempting to transfer the DFS files onto ADFS discs using the *DFSADFS command contained in the ADU Rom. The transfer appears to work ok, but when I try to use the resulting ADFS disc, everytime a program reaches a DRIVE statement i get an error "Not found at <xxx>". In the ADU manual it says that any program written for DFS needs every occurance of DRIVE to be changed to MOUNT before it will work in ADFS. I did try this but it still wouldn't work. So has anyone any ideas of what I could be doing wrong?...!

Norma Lee, BB025

BBC PD closed down last year, and all of the PD discs were kindly donated to Chris at 8BS. Through Chris, you will be able to get hold of the 8-Bit collection and BBC PD collection of discs.

I thought that you and your members might find the information contained in the enclosed leaflet of interest. Greenacre Services are a local firm who specialise in servicing and selling Acorn computers, in particular the original BBC range. I believe they have a contract to maintain and repair computers in Berkshire schools. In the past twelve months I have used them on several occasions and I have found them to be very reliable, they can repair computers and monitors as well as disc drives and printers, and I consider their charges to be very reasonable. For example they repaired my back up printer, a Mannesmann Tally MT160, for £32.60 compared to a figure of £134.00 quoted by Mannesmann Tally, and that was just for the parts.

Perhaps I should make the usual disclaimer at this point and say that I have no connection with the firm other than as a satisfied customer. I have checked with the proprietor, Mick Elliot, and he would be perfectly happy if you decided to

mention his firm in ByteBack. I explained to him that we were a small user group of some 60 or 70 members (it's probably climbed above that by now?) dedicated to using the BBC range of computers.

Ian Bell, Reading, Berks.

Greenacre Services – Beales Lane, Tilehurst, Reading, Berks RG3 5UD. 01734-422422

Thank you for Issue 10. Pleased to see a few pages of letters in there and an interesting seasonal offering. Although the Memory Editor program is a splendid and useful utility to have, I do wonder how many readers have bothered to type it in, and get it to run. I have not yet had the chance, but will in the next day or two. Unfortunately when I have intentions to use the Beeb, I normally end up being distracted and just have that 'one more go' at Elite, Revs, etc. (I still play them!) I hope you had a happy and pleasant Christmas, and have a prosperous New Year.

John Sampson, Leeds

As it happens, Christmas was pleasant: my wife and I were in Los Angeles for a week, followed by seven days in San Francisco! Whilst everyone was holidaying last summer, we were working away until the small hours. The only time we can take a break is over Christmas.

I recently wrote to ByteBack, providing info on recycling old and salvaged Model B's and offer the following in respect of an unusual, misleading and time-wasting fault which appeared on one of my keyboards. The CAPS LOCK key was inoperative with the LED on permanently and characters appearing as a mixture of upper and lower case indicating a possible faulty IC on the main PCB, however a replacement keyboard worked correctly.

Initialy the keyboard to PCB link, CAPS LOCK keyswitch, diodes and resistors on the keyboard were tested and a check made for dry joints and cracked tracks but no fault was found. This left the switch matrix which is controlled by a 74LS251 data multiplexer, 74145 (or 7445) decoder, 74LS163 counter and 74LS30 nand-gate, any of which might be faulty, the first most likely but in the end I removed and replaced them all, with no luck.

Finally a low resistance of around 600Ω was discovered across both the parallel connected SHIFT keys, elimination showing it to be across the left-hand one, the one which probably gets hammered the most!! A resistive switch is most unusual, normally a fault would be an open or short circuit. Measuring resistance can be difficult with IC's and diodes in circuit as the values can be masked by conducting junctions if the voltage used by the ohm-meter is too high, in my case the IC's were removed. Obtaining a replacement keyswitch from a spare board can be made difficult by the fact that the types used may vary with different keyboards and they are not interchangeable.

Changing IC's on some of these boards can be difficult as it is easy for the tracks to peel off if solder-suckers or desoldering braid is used. Using a sharp pair of side-cutters, I cut ACROSS the IC body

between each pair of legs and this cracks the plastic body leaving the legs with bits of innards attached. Each leg is then carefully removed using the tweezers and a small soldering iron with a fine bit. The hole in the PCB is then gently cleared FROM THE SOLDER SIDE with the iron and a pin (I have used a safety pin for years). I then usually fit a DIL socket (to cater for future casualties) and the new IC. These parts are cheap and available from such firms as Maplin's.

N L Smith, Staffordshire

Congratulations on the impressive colour Christmas front to this issue. Rather belatedly I would say that I would support whatever format is most convenient to you but if you did decide on the large sheet could it be printed so that it can be use-friendly to the extent of it being possible to cut it out and assemble it in A5 booklet form. I spent some time fooling around before deciding it was impossible the way round you'd printed it.

Letters in BB10 — I could lend Mrs DM Graham a tape of Scrabble for her to copy. Incidently it does not work on a Master, much to my disappointment. I have not yet worked out how to get it to do so. (I bought a disc version that was alleged to, but didn't so I returned it). If anyone knows how, I would love to hear about it – I believe it is the only version of Scrabble issued for the Beeb (Leisure Genius). I can't help with Cribbage.

It seems to me that the future for the Beeb/Master is going to be in the hands of loyal enthusiasts who can also repair their machines (though even they will depend on a supply of spares). Unfortunately I am not numbered among them – I cannot even use a soldering iron. Articles like that by Mr Smith in Issue 9 could be a great help here.

Those of us who have Masters have the additional problem of needing a regular supply of battery packs. It looks to even my unknowledgable eyes that they would be fairly simply to make up. Could someone provide some instructions for the magazine? (If I can't do it myself I could probably find someone who could).

Has any group put together a list of repairers who are still handling Beebs/Masters? No doubt some BBC Breakers will soon surface (perhaps some already have) to supply second-hand spares.

I believe it is also possible to run a Master without a battery (though of course one would lose the useul clock/calendar facility)

Christina Chesterfield, Truro, Cornwall

It is perfectly possible to operate the Master without batteries, but you are right that you would lose the constant, real-time clock information and all of your default settings (screen mode on boot-up, key repeat rate, port speeds, etc). I know of somebody who uses their Master with a home-made battery pack. I think it as a case of connecting three AA-type batteries in series, but I can't find the details unfortunately.

With respect to the old 'new' format of BB, it made production a little easier, and printing of the pages on the centre section could have been set any way around that

members preferred. However, with the lack of time presently available to me, and the impending greater demand on my time, ByteBack will not be available in any format, I'm sad to say.

I see Murphy hasn't deserted you yet as he has left his mark on your Article Sheet No.1, Side 2, "DELETE UP TO CHARACTER – deletes... and all characters to the *right...*" obviously should be to the left!

Now for a few random thoughts (sorry about that!) on RANDOM SWIZZ!! in same issue. There is no need to use INT with RND as RND(n) always returns an integer if (n)=2 or greater. If (n) is a non-integer, RND will ignore any decimal part and treat (n) as an integer.

Bernard Beeston, Enfield

As always, I am grateful for Your support and advice on matters pertaining to the Beeb! It's useful to have somebody to pick me up on the things I miss!

Enjoyed the Christmas edition of ByteBack. Front page did you credit. Editorial good, enthusiasm and ambitions came across very well. That CE Indexer program is VERY useful. Letters are always very interesting and other people's problems are sometimes yours. My brother, a maker of model steam locomotives for 35 years having kept every railway mag for that period seemed to have the ability to remember letters to the Editor that solved his problems with unerring success despite being 10, 15 or 20 years ago! The Christmas lights program was very interesting. The original idea of that screen for the Melvyn Wright disc was to get the stars featured as a frame, to the introductory wording to move round. I shall try and see if i can use this program. Wonder if the 'double height' print subject matter, might affect it? Ah well! Wait and see.

Frank Jones, North Yorkshire

As you know, 8BS has been invited to the next Acorn User Show at Harrogate on the 6th and 7th of May. 8BS now has 218 members, 653 discs in the pool, 275 packages through the letterbox this year so far! I intend to have a bit of fun at the show this year! Any moral support from ByteBack members would be great!

Chris Richardson, 8BS

IT'S A GAMBLE *cont...*

```
110 REPEAT
120 FLAG=FALSE
130 FOR loop=0 TO 4
140 IF ball(loop)>ball(loop+1) temp
=ball(loop):ball(loop)=ball(loop+1)
:ball(loop+1)=temp:FLAG=TRUE
150 NEXT
160 UNTIL FLAG=FALSE
170 PRINT;ball(0)",";ball(1)",";ba
ll(2)",";ball(3)",";ball(4)",";ball
(5)
180 PRINT
190 END
```

SOFTKEYS

Brian Hawkins

Softkey programs are usually relatively short, but the strings may be allowed to extend to some 240 bytes or more. Used in conjunction with *FX138, which allows one key to call another, quite sophisticated utilities can be provided. (I would like to thank past contributors to Beebug for some of the techniques used in the following examples)

Program 1 - Memory Dump - allows memory to be displayed on a single keypress. The Basic procedure is listed below, followed by the equivalent softkey version.

```
1000 DEFPROCmemdisp:VDU22,3:*FX202,32
1010 PRINT"MEMORY DISPLAY"'STRING$(79,"-")
1020 INPUT"Start address:- "s$:s=EVAL("&"+s$):VDU28,0,24,79,2,14,12
1030 REPEATPRINTRIGHT$("000"+STR$~s,4)SPC4;:k$="":FOR i=0 TO 15
1040 k=i?s:PRINTSPC(2*(-(i=8)));RIGHT$("0"+STR$~k,2)" ";
1050 k=k+(k<32ORk>126)*(k-46):k$=k$+CHR$k:NEXT:PRINTSPC3;k$
1060 s=s+16:UNTIL s=&FFF0:ENDPROC
```

This procedure displays memory from the given address (say &E00) with 16 bytes per line plus the ASCII equivalent where possible. The softkey version using f0 and f3 is below:-

```
600 DEFPROCmemkey
610 *KEY0 MO.3:OS."FX202,32":P."MEMORY DISPLAY"'STRI.79,"-"):I."Start address:- &"s$:s=EVAL("&"+s$):V.28,0,24,79,2,14,21:OS."FX138,0,131"|M
620 *KEY3 V.6,12:REP.P.RI."000"+STR$~s,4)SPC4;:k$="":F.i=0TO15:k=i?s: P.SPC(2*(-(i=8)))RI."0"+STR$~k,2)" ";:k=k+(k<32ORk>126)*(k-46): k$=k$+CHR$k:N.:P.SPC3k$:s=s+16:U.s=&FFF0:ENDIM
```

Note that the character before the final 'M' in each key definition is the 'vertical bar' on the key to the right below BREAK. *FX138,0,131 puts the contents of f3 (key code 131) into the keyboard buffer.

Program 2 - File dump - similar to *DUMP, but screen display as for program 1.

```
1100 DEFPROCfiledisp:VDU22,3:*FX202,48
1110 PRINT"FILE DISPLAY"'STRING$(79,"-"):Input"Filename:- "F$
1120 G%=OPENUPF$:PRINT"File open (use *CLOSE to close):*FX202,32
1130 INPUT'"Start address:- &"s$:s=EVAL("&"+s$)
1140 PTR#G%=s:VDU28,0,24,79,4,14,12
1150 REPEATPRINTRIGHT$("000"+STR$~s,4)SPC4;:k$="":FOR i=0 TO 15
1160 k=BGET#G%:PRINTSPC(2*(-(i=8)));RIGHT$("0"+STR$~k,2)" ";
1170 k=k+(k<32ORk>126)*(k-46):k$=k$+CHR$k:NEXT:PRINTSPC3;k$
1180 s=s+16:UNTIL EOF#G%:CLOSE#G%:ENDPROC
```

The softkey version follows:-

```
700 DEFPROCfilekey
710 *KEY1 MO.3:OS."FX202,48:P."FILE DISPLAY"'STRI.79,"-"): I."Filename:-"F$:G%=OPENUPF$:P."File open (f2 to close)": OS."FX202,32"*I."Start address:- &"s$:s=EVAL("&"+s$):PTR#G%=s:V.28,0,24,79,4,14,21:OS."FX138,0,132"|M
720 *KEY4V.6,12:REP.P.RI."000"+STR$~s,4)SPC4;:k$="":F.1=0TO15: k=B.#G%.SPC(2*(-(i=8)))RI."0"+STR$~k,2)"
```

```
";: k=k+(k<32ORk>126)*(k-46):k$=k$+C
HR$k:N.:P.SPC3k$: s=s+16:U.EOF#G%IM
 730 *KEY2 OS."CLOSE"IM
 740 ENDPROC
```

Softkey Buffers:-

In the 'B' these are located at &B00-&BFF with the first 17 bytes pointers to the softkey strings. The 'Master' stores this data in 'ANDY' at &8000-&83FF, using the first 35 bytes as pointers. If the total string space is exceeded, a 'Bad key' error is signalled. The 'B' buffer can be inspected by 'peeking' using PRINT ?&B00 etc. For the Master, 'ANDY' must be 'paged in' first, by setting bit7 of 'ROMSEL' at &FE30. This usually contains &0C, the 'C' referring to the BASIC ROM number of 12. To set b7 requires ?&FE30=&8C. The short routine below will display the contents of 'ANDY' from &8000-&83FF:-

```
 900 DEFPROCandy
 910 VDU22,3:FOR I%=0 TO &3FF:REM Set
Mode3
 920 ?&FE30=&8C:?&70=I%?&8000:?&FE30=
&0C:REM put byte in &70
 930 IF I%<&22 PRINT;~?&70" ";:NEXT:R
EM Pointer bytes
 940 k=?&70:IF k>31 AND k<127 PRINTCH
R$k" ";:NEXT:REM Text bytes
 950 PRINT;~k" ";:NEXT:REM Not text
bytes
 960 ENDPROC
```

These routines need very careful typing in! The program should not be 'RUN' since it only consists of procedures! The softkey routines can however be set up by calling say: PROCmemkey <RET> & PROCfilekey <RET>, which will allow key f0 to provide a MEMORY DISPLAY, f1 a FILE DISPLAY, with f2 providing file closure. I am working on a disc sector display, but it is difficult to produce a single routine for both DFS and ADFS since they make use of OSWORD &7F and OSWORD &72 respectively. In addition, DFS requires the drive number, track number and sector number, while ADFS only needs the 'absolute' sector number. At the moment the solution appears to be to have a separate routine for each filing system.

Two miscellaneous routines:-

```
 1300 DEFPROCcurs(n):LOCAL A%,x:A%=13
5:!&4FA=USR&FFF4
 1310 x=20+76*n:IF n=2 x=103-11*(?&4F
C=7)
 1320 VDU 23;10,x,0,0,0,0,0,0:ENDPROC
```

This provides cursor control in any Mode:- n=0: none, n=1: large flashing for data input, n=2: normal A%=135:!&4FA=USR&FFF$ uses OSBYTE 135 to obtain the Mode in use, returning 0-7 in &4FC. x=103 suits all modes except 7, the 11*(?&4FC=7) corrects x to 114 for this latter Mode.

```
 1400 DEFPROCin(n):PROCcurs(1):K$=""
:REPEATK%=GET
 1410 IF K%=127 AND K$<>"" VDU8,32,8
:K$=LEFT$(K$,LENK$-1) ELSEIF K%>31
AND K%<127 AND LENK$<>n K$=K$+CHR$K%
:VDUK%
 1420 UNTIL K%=13:PROCcurs(0):ENDPROC
```

This simple input routine seems sufficient for most needs. It limits the characters to the range ASCII 32 to 126, and allows a string of up to n characters. Input is completed by <RET>, and the delete functions normally.

EXTRAORDINARY

Andrew Bennet

Acorn's 8-bit machines hold little mystery for me. During nine years use, I've explored many of their facets. So, I was surprised to find something that I hadn't noticed before.

It was last Easter, and I was engaged in some unmemorable tinkering. For some reason, I tried BASIC 2's power operator with particularly large values. It was some thing like 2^100. The answer was correct, but a noticable pause for calculation caught my attention.

At first, I blamed a screen update anomaly, but further tests had a similar lathargy. I began to investigate.

Knowing that power functions use formula 1 for evaluation, I quickly found BASIC's EXP function to be the cause. A test program revealed a clear decrease in BASIC 2's performance as EXP's argument becomes larger. The same program went through Accelerator, Inter-Base, and OPL on a Psion 3a. None showed BASIC's linear increase, times were more or less constant. The question is, why? What follows is my guess.

Most A-level math's text books give Maclaurin's series, formula 2, as EXP's solution. In theory, its result is incomplete outside of infitity, but, in practice, algorithms only use the terms significant enough to affect the floating point resu lt. Unfortunately, for 2^100 this is nearly 130 terms. BASIC 2 appears to calculate them all, whereas the others don't.

Consider formula 3, it shows a rearrangement drawing out a power of two. By ensuring a is small, EXP is guaranteed to execute quickly. The final result is computed by repeatedly squaring the answer b times. The listing's function FNP does this , and for larger results is quicker than BASIC's own power routine.

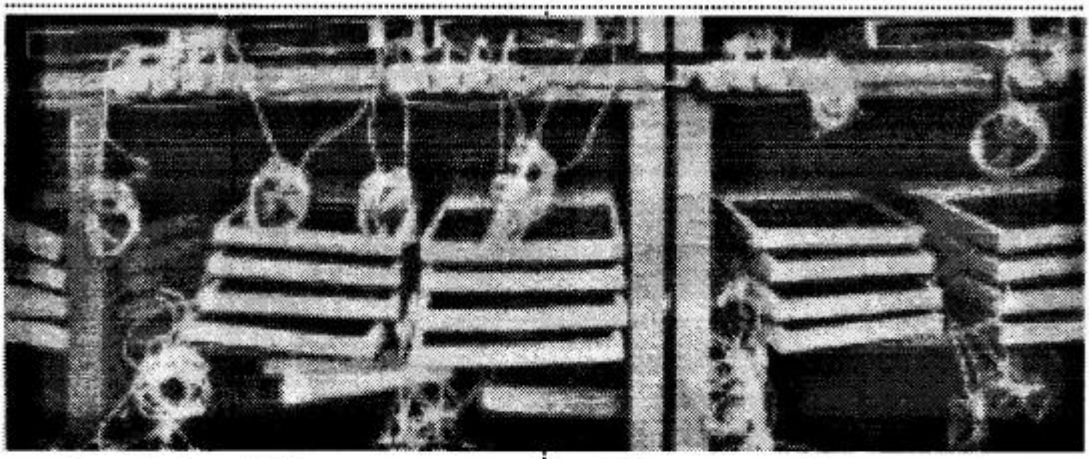
The program will repeatedly raise 2 to values between 10 and 120, shown in the first column. The times taken by BASIC are in the second column, and the function's time in the last.

I suspect this feature is rectified in BASIC 4, but for BASIC 2 users it's something worth trying. Like me, you may find it quite extroadinary.

```
10REM Fast BASIC powers
20REM by Andrew Bennett
30:
40N%=0
50REPEAT
60TIME=0
70FOR J%=0 TO 9
80A=2^N%
90NEXT
100A%=TIME
110TIME=0
120FOR J%=0 TO 9
130A=FNP(2,N%)
140NEXT
150PRINT N%,A%,TIME
160N%=N%+10
170UNTIL N%=130
180END
```

USING YOUR COMPUTER

Trevor Crapper



In this article I would like to give you some ideas of the uses that you can put your computer based on my own experience. Most of the people reading this will use their computers to some work with, i.e., writing letters, keeping household accounts, artistic work drawing or designing things, and the list goes on right down to doing the pools. The last thing they will do is run a game unless it's an old favourite and one needs to relax from a hard stint.

Before going on I feel it necessary to say something here about getting enjoyment from using your computer. If using your computer to assist in some project becomes a chore then don't use it. There is no point if you have to drive yourself, it has to be unobtrusive, an extra tool something that will help you get things done simpler.

I play correspondence chess and last year an opportunity arose for me to give something back to my sport, the position of Team Captain for International Matches became vacant and I applied for the post and got it, the fact that I was the only applicant had nothing to do with the outcome, it is voluntary and there are no perks! I had run a tournament for over 10 years in the 70s and 80s with a different club and so had an idea of what was involved. In those days it was all paper and pencil, home computers had not arrived on the scene. Actually the new tournament is far more involved but that is another story.

Softkeys...

```
 190:
 200DEF FNP(A,B)
 210LOCAL J%,I%,T
 220N=LNA*B
 230I%=0
 240IF ABS(N)>=8 N=N/2:I%=I%+1:GOTO
240
 250T=EXPN
 260IF I%>0 FOR J%=1 TO I%:T=T*T:NE
XT
 270=T
```

Using your computer...

Is there any advantage to be gained from using a computer since things went along quite smoothly in the old days without one, the answer is yes. In order to give some idea of what use a computer can be let's look first of all at what the job is about. In my case writing letters, preparing and printing pairing notices. Something to take care of my expenses, this is a major item since there is a lot of postage plus stationery to be bought, used and accounted for. The number of player involved in a match can be over 100 and besides this there are players writing in for matches fairly regularly and these have to be kept on a list.

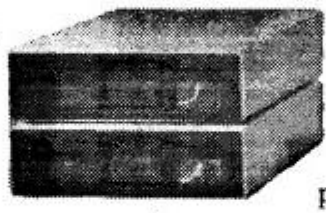
So there you have it a very brief account of what the job entails, your job will certainly be different but may run along similar lines in how the computer could be used. It is not my intention to give lessons on how to use the programs mentioned, this will be left to the experts.

In the first place taking letter writing and pairing notices together you need the services of a good wordprocessor, I use Inter-Word for this purpose, it is extremely user friendly when it comes to the pairing notices which uses different fonts and styles. Using the ruler to it's full advantage does require some thought especially if you are changing font size, experiment is the only solution but, don't become frustrated if it does not work out right first time. The method I use is to draw a rough draft on paper and try and get some idea of the page width and choose a font size that will fit. Look it on the screen before you commit it to paper you may have to alter the page width a few times before the writing fits in properly. You may also want to underline something or emphasize it or do both, remember to switch it on and off. This may have to be done using the built-in facilities and once again remember to program things in and out. Underlining can be a pain and a simple way round this is to do it line by line rather than continuous.

The next thing I want is a good database so that all the players names, addresses, their playing strength, the number of matches requested and the those they are playing in currently along with any scores can all be kept ready for instant recall. Databases seem to be the mystery program to a lot of people, me included. You look at the jargon and think this is not for me, what's it all about, I'll never learn that. Databases are used to hold records, each one of the players is an individual record in mine, my record is there as well. The nature of the records held is almost limitless almost everything under the sun can be catalogued in this way. Some preparation is needed before you start to your database, obviously what you intend to store records on and why, how do you want to view the database once it is up and running. Here again paper and pencil to draw a rough draft copy of a record is the best way to start. You will have to look at the layout and decide for yourself how you want it to look on the screen. I am using the Inter-Base Card Index currently, this

is a very simple to program provided you follow the instuctions, you create a database first and this is where the preparation will come in very useful because you should know how my lines your database is going to take up. Each of the lines is a Field and a Record is the number of Fields used to store the information on. In addition I have also just fathomed ViewStore out and will compare the two databases for ease of use before deciding which one to use.

The other utility I use regularly is ViewSheet which is part of the Master's in-built suite of roms. ViewSheet is a spreadsheet, this means that space for inputting data is spread over a large area. The system uses letters and numbers in combination to denote individual spaces, these spaces are known as boxes and this is where you put your data. Box A1 is at the top left hand corner, box B1 is the next one along and so on. Under box A1 is box A2, box A3 etc., and in order to see the whole of the spreadsheet it is necessary to scroll both down and along. A spreadsheet is used basically for containing numbers and you can perform mathematical equations on these numbers by input the box number then the equation '+ * - /' and then the next box number followed by the next equation and so on. This is a very simple process especially if like me your maths aren't very good. A spreadsheet can be used as a database with the additional feature of being able to input and work on numbers and this really does open out its potential. Household accounts, even small business accounts are not beyond its scope, Club treasurers would find this a boon. Of all the applications I find this one is the easiest to use and ViewSheet is very user friendly just refer to your Welcome Manual for some very good information, this is all I need to keep me going.



My final offering is Desk Top Publishing and I used the AMX Stop Press for this. Desk top publishing can be used to produce such things as newspapers, magazines, posters, notices, flysheets, handouts etc. I have to admit that my involvement was only on the fringe and there were no photos or drawings. I used to produce a newletter on a regular if somewhat erratic basis along with notices and entry forms. Up until last November I was Chairman of a local radio controlled model racing car club and had produced these for the club. It is a good idea to draw a rough draft of what your finished article will look like. An A4 poster or notice is very easy to produce and you can use various fonts to give them eye catching highlights.

This article is just a pointer and I hope it may help those of you who have these utilities but don't know what to use them for, or you may like to buy one, then see Martin Pickering's advert. Do have a go, it can be great fun and it certain helps you to get to know your computer and to gain more confidence in using it.

INFLATION

Frank Iveson

BYTE BACK

In searching for an application to prove the use of Column first, Row second principle (as applied in PRINTTAB) in an array, as opposed to the way it is usually illustrated in the programming guides, i.e. Row first Column second, I recalled a newspaper clipping I had extracted a few years ago showing data corresponding to inflation occurring between 1891 and 1990, based on the state of the £1 in 1914. I have updated this to 1994 and I feel the product of my labours are worth passing on as it makes sense of the expression 'in real terms' and allows one to check on what the politicians say in this respect.

It is both interesting and informative to consider the effects of inflation, and more than a little annoying that you didn't have the foresight to anticipate it to good effect. For example, did you know that the average house price in 1953 of £2,013 is the equivalent in present terms (inflation adjusted) of £27,562 but is actually £59,800 (aren't we being ripped off) or, that a Beeb computer purchased for £500 is in present terms equivalent to £846... (but second-hand costs nearer to £50. Anyone want to buy a BEEB?) Conversely, a house which today costs £90,000 in 1950 values would be £5553. See the difference to £2,013 of 1950. Makes you think, doesn't it?

```
10REM INFLATN
20REM Inflation calculator
30REM (c) Frank Iveson
40REM April 1994
50ON ERROR IF ERR=17 VDU22,7:FOR i%=
11 TO 12:PRINTTAB(7,i%)CHR$141;CHR$1
31"Do you want to quit?":NEXT:ans$=F
Nget("YN"):IF ans$="Y" GOTO 750:ELSE
IF ans$="N" CLEAR:GOTO 60 ELSE REPOR
T:PRINT" at line "ERL:PROCpause(300)
:STOP
60MODE7
70DIM YR(104,2) :REM ***** All the
'*'s are associated with number of
years
80:REM ***** Presently set for 104
years 1891-1994 ******
90FOR C=1 TO 104
100FOR R=1 TO 2
110READ YR(C,R)
120NEXT R
130NEXT C
140CLS
150PROCtitle(CHR$135+"Inflation")
160PRINTTAB(6,3)CHR$131"(Range is 18
91 - 1994)"
170REPEAT
180E1=FNgetno(0,6,"Enter earlier yea
r:","0123456789",4)
190UNTIL a$=CHR$13
200FOR S=1 TO 104
210IF E1 = YR(S,1) THEN E2 = YR(S,2)
220NEXT
230IF E1>=1995 OR E1<1891 PRINTTAB(6
,8)CHR$134"OUT OF RANGE":PROCpause(2
00):PRINTTAB(6,8)SPC20:GOTO 170
240@%=10
250PRINTTAB(0,8)"Corresponding fact
or for ";E1;" is: ";:@%=&20106:PRINT
;E2
260@%=10
270REPEAT
```

```
280L1=FNgetno(0,10,"Enter later yea
r:  ","0123456789",4)
290UNTIL a$=CHR$13
300FOR S=1 TO 104
310IF L1 = YR(S,1) THEN L2 = YR(S,2)
320NEXT
330IF L1>=1995 OR L1<1891 PRINTTAB(6
,12)CHR$134"OUT OF
RANGE":PROCpause(200):PRINTTAB(6,12)
SPC20:GOTO 270
340IF E1>L1 PRINTTAB(0,12)CHR$134"Ea
rlier is greater than later!":PRINTT
AB(0,13)CHR$134"Please re-enter.":PR
OCpause(150):PRINTTAB(0,12)SPC39:PRI
NTTAB(0,13)SPC39:GOTO 140
350@%=10
360PRINTTAB(0,12)"Corresponding fact
or for ";L1;" is: ";:@%=&20106:PRINT
;L2
370@%=10
380PRINTTAB(0,14)"Earlier or later
value (E/L) ?"
390A$=FNget("EL")
400REPEAT
410A=FNgetno(0,16,"Enter amount for
conversion:",".0123456789",7)
420UNTIL a$=CHR$13
430:
440REM  calculations then new screen
450CLS
460PRINTTAB(5,2)"Earlier year:  ";E1
470PRINTTAB(5,3)"Later year:    ";L1
480@%=&20206:PRINTTAB(2,5)CHR$134"Am
ount for conversion: £";A
490@%=10
500PRINT' STRING$(38,"-")
510IF A$="L" GOTO 620
520IF A$="E" INF=((L2-E2)/E2)*100
530EAR1=A*E2/L2
540EAR2=A*L2/E2
550PRINTTAB(2,9)CHR$134"To earlier
year:       ";E1
560@%=&20206:PRINTTAB(2,11)CHR$134"
Inflation change is: ";INF"%"
570PRINTTAB(2,13)"Earlier year purch
asing power,"
580PRINTTAB(2,14)"of £";A;" now,
was: £";EAR1
590PRINTTAB(2,16)"Earlier year value
equivalent,"
600PRINTTAB(2,17)"of £";A;" now,
was: £";EAR2
610IF A$="E" GOTO 720
620IF A$="L" INF=((E2-L2)/E2)*100
630LAT1=A*L2/E2
640LAT2=A*E2/L2
650PRINTTAB(2,9)CHR$134"To later
year:       ";L1
660@%=&20206:PRINTTAB(2,11)CHR$134"I
nflation change is: ";INF"%"
670PRINTTAB(2,13)"Later year purchas
ing power,"
680PRINTTAB(2,14)"of £";A;" then, is
: £";LAT1
690PRINTTAB(2,16)"Later year value e
quivalent,"
700PRINTTAB(2,17)"of £";A;" then, is
: £";LAT2
710@%=10
720PRINTTAB(3,20)CHR$131"Do you want
another go (Y/N)"
730A$=FNget("YN")
740IF A$="Y" GOTO 150
750VDU22,7
760@%=10
770END
780:
790DEF PROCpause(x%)
800pause=TIME+x%
810REPEAT UNTIL TIME>=pause
820ENDPROC
830:
840DATA 1891,111.0,1892,111.0,1893,
114.4,1894,116.8,1895,122.0,1896,123
.3,1897,118.1,1898,119.4,1899,118.1,
1900,114.4
850DATA 1901,113.3,1902,112.1,1903,
111.0,1904,109.9,1905,111.0,1906,11
```

```
1.0,1907,107.8,1908,104.7,1909,104.7
,1910,103.7

860DATA 1911,102.8,1912,99.1,1913,9
9.1,1914,100.0,1915,81.0,1916,68.5,1
917,56.6,1918,49.1,1919,46.4,1920,40
.1

870DATA 1921,44.2,1922,54.4,1923,57.
2,1924,56.9,1925,56,1926,58.1,1927,5
9.5,1928,60.0,1929,61.0,1930,63.1

880DATA 1931,67.7,1932,69.4,1933,71.
2,1934,70.7,1935,69.8,1936,67.7,1937
,64.5,1938,63.8,1939,58.0,1940,56.0

890DATA 1941,53.0,1942,50.0,1943,48.
0,1944,45.0,1945,41.0,1946,38.3,1947
,35.8,1948,33.3,1949,32.3,1950,31.6

900DATA 1951,28.7,1952,27.4,1953,26.
7,1954,26.2,1955,25.4,1956,24.2,1957
,23.5,1958,22.8,1959,22.6,1960,22.3

910DATA 1961,21.7,1962,20.9,1963,20.
6,1964,19.9,1965,19.0,1966,18.3,1967
,17.8,1968,17.0,1969,16.2,1970,15.2

920DATA 1971,13.9,1972,13.0,1973,11.
9,1974,10.2,1975,8.2,1976,7.1,1977,6
.1,1978,5.6,1979,5.0,1980,4.2

930DATA 1981,3.8,1982,3.5,1983,3.3,1
984,3.1,1985,3.0,1986,2.9,1987,2.8,1
988,2.6,1989,2.4,1990,2.2

940DATA 1991,2.06,1992,1.98,1993,1.9
5,1994,1.85,1995,,1996,,1997,,1998,,
1999,,2000

950REM Source of 1891-1990 is the
Central Statistical Office

960REM Note: The years 1939-1945
inclusive are interpolated as no
stats avail.

970REM The years 1991-1994 have been
calculated from BBC2 CEEFAX RPI%

980REM increases on previous year
table (P281/2of5)

990:

1000DEF FNgetno(x%,y%,prompt$,allow
ed$,l%)

1010LOCAL in$:in$=" "

1020allowed$=allowed$+CHR$13+CHR$127

1030REPEAT:REPEAT

1040PRINTTAB(x%,y%)SPC(40)TAB(x%,y%
) ;prompt$+" ";in$;

1050*FX15,1

1060a$=GET$

1070REM  respond only to valid chara
cters

1080UNTIL INSTR(allowed$,a$)

1090REM only one decimal point

1100IF a$="." THEN IF INSTR(in$,".")
<>0 a$=""

1110REM add to string if not deleted
or CR

1120IF a$<>CHR$127 AND a$<>CHR$13 in
$=in$+a$

1130REM respond to delete

1140IF a$=CHR$127 in$=LEFT$(in$,LEN(
in$)-1):IF in$="" in$=" "

1150REM limit size of number

1160IF INSTR(in$,".")<>0:IF LEN(in$)
>l% THEN in$=LEFT$(in$,l%+2)

1170IF INSTR(in$,".")=0 THEN in$=LEF
T$(in$,l%+1)

1180REM restrict decimal places to
<= 2

1190IF INSTR(in$,".")<>0:IF LEN(in$)
-INSTR(in$,".")>2 THEN in$=LEFT$(in$
,LEN(in$)-1)

1200UNTIL a$=CHR$13

1210=VAL(in$)

1220:

1230DEF PROCtitle(t$)

1240VDU 26,12

1250FOR i%=1 TO 2

1260PRINTTAB(0,i%)CHR$132CHR$157SPC(
16-LEN(t$)/2)CHR$141;t$

1270NEXT

1280VDU 28,1,24,39,3

1290ENDPROC

1300:

1310DEF FNget(b$)

1320REPEAT a$=GET$:UNTIL
INSTR(b$,a$)

1330=a$
```

HINTS AND TIPS

a mixture of little nuggets

1. BASIC EDITING

If you are editing a BASIC program, you can find out what is the last line number in the program as follows: Hold down SHIFT and CTRL together, press ESCAPE twice in succession, and finally release SHIFT and CTRL.

This even works on an Archimedes/A3000!

2. LISTINGS

To stop listings and any other screen output* whizzing past too quickly, hold <SHIFT> and <CTRL> down together. Alternatively, from the BASIC '>' prompt, press <CTRL><N> to engage paged mode, and press <SHIFT> to continue scrolling. Cancel with <CTRL><O>. This also works fine on an Archimedes.

3. FILLING UP STRINGS

It is sometimes necessary to fill-up or 'pad' out strings to a fixed length. For example, you may want numbers to print out as 0045, 0001 etc., or you may want to add leading or trailing spaces onto character strings. The first example gives the obvious method, whereas the second shows a more elegant method which avoids the use of an IF statement. The third example shows a similar method in the form of a Function, with the addition that you can specify the amount of padding and the padding character to be used - it doesn't have to be a number.

```
30 IF LEN(X$)<4 THEN X$="0"+X$:GOTO 30
```

```
30 X$=STRING$(4-LEN(X$),"0")+X$

10 INPUT"Enter up to 5 digits "A$
20 A$=FNpad(A$,5,"0"):PRINT A$:END
30 :
100
DEFFNpad(st$,len%,char$):=STRING$
(len%-LEN(st$),char$)+st$
```

4. RANDOMIZE

To 'scramble' the Beeb's random number generator properly, you can simulate the RANDOMIZE function of some other BASICs. Incorporate the dummy expression randomize=RND(-TIME) at the beginning of your program. This also works on an Arc/A3000.

5. SELF-VALIDATING INPUT

Try incorporating this sort of input routine into your own program. There is nothing special about the particular line numbers used.

```
1000 PRINT"Do you want another game?"
1010 ON INSTR("YyNn",GET$) GOTO 102
0,1020,1030,1030 ELSE 1010
1020 REM Back to start of program.
1030 END
```

6. SHORTENED 'IF' STATEMENT

In most cases, the statement IF A<>0 THEN etc. can be shortened to just IF A etc. The "<>0" is implied, provided A can only be zero, or +1 or more, or -1 or less. In the 'before' and 'after' example below, note

the space after the variable "A" in the second version. This space would be unnecessary if the variable were "A%" instead.

```
100 IF A<>0 THEN G=5:GOTO 70

100 IF A G=5:GOTO 70
```

(Note the space after the A!)

7. SIMPLE BLEEP

If you want a short 'bleep' in your program, you can use VDU7 or PRINT CHR$(7) instead of a SOUND statement. VDU7,7 etc would give a longer bleep.

8. VDU7 BLEEP

You can alter the nature of the VDU7 'bleep' as follows. *FX210,1 turns it off, and *FX210 turns it on again. You can alter the pitch with *FX213,P where "P" is a number from 0 to 255, (default value is about 100). Similarly, you can alter the duration with *FX214,D (default value about 7). You can change sound channel or 'voice' with *FX211,N where "N" is 0 to 3, (normally 3). Thus, if you try *FX211,0 you will get a sort of explosion noise, and *FX211,3 will restore it to a bleep. You can test the bleep by pressing <Ctrl><G>; quicker than typing VDU7.

9. KEY-PRESSED CHECK

You can check to see if a specific key is being pressed at a particular instant with a negative INKEY statement. For example, you can test the space-bar with IF INKEY(-99) THEN... However, if you want to see if ANY key is being pressed, (though not the red keys or <Ctrl> or <Shift>), then use this command. It works in the opposite sense to the other negative INKEY commands, hence the inclusion of the word "NOT". IF NOT INKEY(-129) THEN...

10. INPUT LINE

An ordinary INPUT statement will strip any leading spaces off strings, and will not accept commas within a string. The latter is because you can use commas to separate your replies, instead of pressing <Return> each time, eg BLOGGS,45,MALE <Return>. You are reminded that the alternative command INPUTLINE will accept commas and leading spaces. You can easily strip the leading spaces off afterwards, if they are a problem.

11. STRING INPUT FUNCTION

This Function is an alternative to using a conventional INPUT statement for strings. You specify the minimum and maximum number of characters in the string, and the Function does the rest. The example given is for 2 and 6 characters respectively, whereas an INPUT statement is effectively for 0 and 255. You will find that you cannot press <Return> until you have typed in the minimum number of characters, and that you cannot type more than the maximum. Full use of the <Delete> and <Copy> keys is allowed. This Function is quite tricky to type in without errors, so take care. I've only put spaces in for clarity, and you can omit them

all except immediately before the word "ELSE". If you want the cursor to remain on the same line after <Return> is pressed on a valid string, then omit the final PRINT: in line 140, so that you have just ELSE =I$.

```
10 PRINT"Enter Name ";:name$=FNinp(2
,6):GOTO 10
100 DEFFNinp(min%,max%):LOCALG%,G$,I
$:*FX21,0
110 PRINT G$;
120 G$=GET$:G%=ASC(G$)
130 IF(LEN(I$)=max% AND G%<>13 AND G
%<>127)OR(((LEN(I$)=0 AND(G%<33 OR G
%>126))OR(LEN(I$)<min% AND G%=13))AN
D(min%>0 OR G%<>13))THEN VDU7:GOTO 1
20
140 IF G%>31 AND G%<127 THEN I$=I$+G
$:GOTO 110 ELSE IF G%=127 THEN I$=LE
FT$(I$,LEN(I$)-1):GOTO 110 ELSE IF G
%<>13 THEN VDU7:GOTO 120 ELSE PRINT:
=I$
```

12. NUMBER INPUT FUNCTION

This is a similar sort of Function for validating numbers which may be more than just a single digit. If the number entered doesn't fall within the required limits, the incorrect entry is erased ready for another try. If you are only interested in integers, then alter the variables figure, min and max by adding a "%" sign on the end. Wierd things happen if you muck about with the cursor editing keys, but you can easily temporarily disable them with *FX4,1 or *FX4,2. Like the previous Function, it really pays off in larger programs, as it saves you having to write separate validation routines for each and every input. You could save the Function with the *SPOOL facility, and *EXEC it into new programs as required. Again, you can omit spaces if you want.

```
10 CLS:PRINTTAB(7,10)"Enter Width ";
:width=FNnumber(18.2,65.7)
20 END
100 DEFFNnumber(min,max):LOCAL P%,V%
,figure
110 P%=POS:V%=VPOS:PRINT:REPEAT REPE
AT VDU127:UNTIL POS=P% AND VPOS=V%:I
NPUT""figure:UNTIL figure>=min AND f
igure<=max: =figure
```

13. STATUS FUNCTIONS

The first is a useful Function which can be used to test the status of various aspects of the VDU drivers. For example, to check if the screen is in paged mode, then use IF FNstatus(2)=TRUE THEN... To check whether there is a text window defined, then use IF FNstatus(3)=TRUE THEN..., and so on. You can achieve the same result, though not in a proper Tube-compatible way, by Peeking location &D0. Indeed, the only way I know of disabling scrolling, (bit 1), is to directly Poke &D0 with ?&D0=?&D0 OR 2. (Enable it again with ?&D0=?&D0 AND 253.) The second Function returns the current graphic Mode number, and the third Function returns the ASCII code of the character at the present cursor position. You can move the cursor to the required position with PRINTTAB(x,y); or VDU31,x,y.

```
100 REM *Returns TRUE if Bit is Set*
110 :
120 REM 0-Printer enabled by VDU2
130 REM 1-Scrolling disabled
140 REM 2-Page Mode enabled by VDU14
150 REM 3-Text Window defined by VDU28
```

Continued on back page

CLASSIFIEDS

BBC Micro with disc interface, speech and econet. Full working order. Bargain at £15. Tel 01223 503382, evenings only.

BBC B, 32kRam, 40 track 51/2" disc drive, Cub monitor, bridge, manuals, some software, all leads, excellent condition, £150. Tel 01254 706267.

BBC Books, 9 books for all BBC's will sell seperately and post, phone for details, £10 lot, Portsmouth. Tel 01705 787669.

BBC Micro, Microvitec 1420 colour monitor (Hi-res), Cumana double drive, Viglen single drive, modem, software, may split, £120 Watford. Tel 0923 237526.

BBC model B, 40/80 disc drive plus tape deck, colour monitor, £55. Tel 0161 969 9894.

Out of print computer books, mainly BBC, ask for list, Acorn User, Micro User, earliest issues available, Smith, 9 Browning Close, Daventry, NN11 5LE. Tel 01327 702095.

BBC Master plus, external hard disc drive, also 51/4" setup disc for floppy drive, £100 + £8 p&p. Tel 0121 777 8808.

BBC Master, drive, software, books, cassette, etc, £130. Ring Tony. Tel 0181 697 4959.

Almost whole collections of Acorn User and BBC Micro User for sale, £30each set plus p&p. Tel 01992 652066 evenings.

Hints and Tips...

```
160 REM 4-Not used
170 REM 5-Text/Graphics cursors
joined by VDU5
180 REM 6-Edit cursor in use
190 REM 7-VDU drivers disabled by
VDU21
200 :
210 DEF FNstatus(bit%):LOCAL A%
220 A%=&75:=-(((USR&FFF4 AND &FF00
)DIV&100)AND2^bit%)DIV2^bit%


100 REM ** Returns Graphic Mode **
110 :
120 DEF FNmode:LOCAL A%
130 A%=&87:=(USR&FFF4 AND &FF0000)
DIV &10000
```

```
100 REM ** Returns Character at
110 REM flashing cursor position
120 :
130 DEF FNchar:LOCAL A%
140 A%=&87:=(USR&FFF4 AND &FF00)DI V
&100
```

16. PROGRAM SIZE

To see how long your BASIC program, excluding variable space etc., type PRINT TOP-PAGE <Return>, or PRINT~TOP-PAGE <Return> if you want it in Hex. This also works on an Arc, with the addition that you can use END-PAGE if you want to include any variable space used by the program after it has been run.

Mar 21, 1995 Membership No. BB023

Dear Chris

I am sorry to have to say that I am unable to continue putting ByteBack together. I have taken a lot of time to think about it, in fact as far back as last summer. I simply don't have the time anymore (I didn't really have all that much to start with!)
Any surplus subscription fees have been refunded (enclosed cheque). I have mentioned in this last issue of an alternative supply of BBC support. If you intend to continue using your BBC, you will find a great number of like-minded people over at 8-Bit Software.

Regards,

Paul

ByteBack, 33 King Henry's Mews, Enfield Lock, Middlesex EN3 6JS